The Visual Computer manuscript No.
(will be inserted by the editor)

Lakin Wecker · Faramarz Samavati · Marina Gavrilova

Contextual Void Patching for Digital Elevation Model

Abstract Digital terrain models can be created by gathering a set of measurements from geometric objects. For various reasons, these models may be incomplete such that fail to meet the requirements defined by their potential applications.

In this work, we develop a novel multiresolution approach to repair the voids commonly found in digital elevation models (DEM). We use the overall shape and structure of the surrounding terrain to build a smooth patch for the void. Then, using a multiresolution obtaining from reverse Chaikin subdivision, we extract the low scale characteristics from the surrounding terrain and apply them to the smooth patch. The results demonstrate that our approach is effective at synthesizing models with realistic characteristics.

Keywords Multiresolution · Terrain · Void Patching · Context Sensitive

1 Introduction

In this paper we introduce a novel technique for patching digital elevation models. Digital elevation models (DEM) are used for terrain representation. DEM data is usually gathered through an automatic sampling of the elevation. The automated gathering process sometimes fails to obtain the elevation data in some areas. We refer to these incomplete areas as *voids*.

DEMs are common digital models used for terrain representation. They consist of an evenly spaced two dimensional grid of values where each value represents the elevation of the terrain at that location. One of the main reasons for their ubiquity is due to their simple structure which requires less storage than other model types.

This work provides a novel context sensitive method for patching the voids found in DEMs. Specifically, we patch the DEMs provided by the Shuttle Radar Topography Mission (SRTM), flown in February 2000 [22]. The goal is to produce a patch with structure and characteristics similar to the surrounding terrain. An intuitive approach is to analyze the samples surrounding the *void* and find those that will provide the most useful information. To determine the usefulness of a real sample, we define *relevance* as: the closer a real sample is to the void the more *relevant* it is. *Relevance* is central to this work and will shape many of the algorithms discussed in this paper.

Our method consists of three steps. First, we must identify each void. Second, we use the overall shape and structure of the surrounding terrain to build a smooth patch for the void. Third, using a multiresolution representation, we extract the low scale characteristics from the surrounding terrain and apply them to the smooth patch.

In the rest of this paper we will cover the approach in more detail. After a brief overview of related work in the Section 2, we discuss data format in Section 3. We then discuss the method in detail in Section 4. Finally, we provide a discussion of the results and draw some conclusions in Section 5.

2 Related Work

In the computer graphics area, the general inconsistencies in automatically scanned data pose an ubiquitous problem for data processing [8]. Due to the varied nature of the data and the voids contained therein, the approaches are also significantly varied. Methods on repairing voids by smooth patches have been proposed for polyhedron[2], 3D surfaces[10] and 3D meshes [20,18].

Image completion and texture inpainting are two methods that focus on patching the missing portions of images that can occur if the image is damaged or some large foreground object has been removed. The surrounding

Lakin Wecker, Faramarz Samavati, Marina Gavrilova
2500 University Drive NW
Calgary, Alberta, Canada T2N 1N4
Tel: 403-220-5110
E-mail: {weckerl,samavati,marina}@cpsc.ucalgary.ca

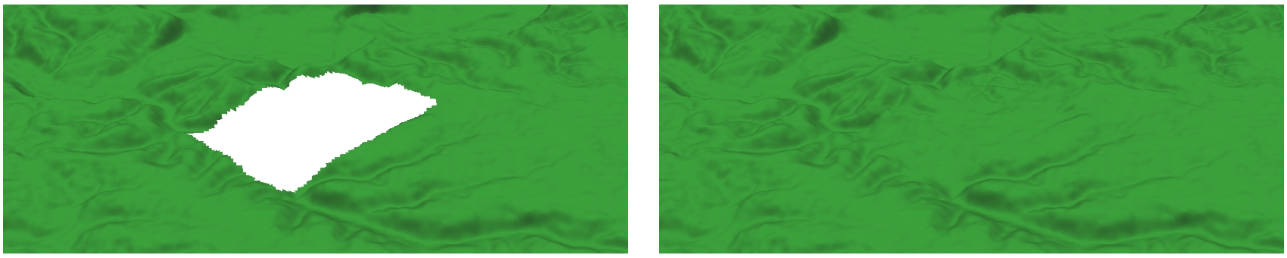


Fig. 1 The left figure illustrates a void. The right figure illustrates the results of our method.

areas of the image are used as training for the inpainting methods, followed by a filling phase that attempts to plausibly patch the missing areas [11,16]. Extensions to these techniques include using inpainting methods on surface models [23].

Over the past decade, researchers have had significant interest in texture synthesis and texture transfer. In texture synthesis, the methods vary from a pyramid-algorithms [5] to Markov random fields [13,25]. Additional techniques use patch based sampling, [17], and some are real time approaches, [25,17]. Improvements have been made in domain specific areas, [1] and further improvements are made in [26].

Texture transfer uses two sample input images and produces a new image that contains similar features to both input textures. Typically, one of the input images provides the texture and the other provides the features for the output image. Approaches to texture transfer and texture synthesis include Wang tiles, [9], image quilting [12], and image analogies [15].

Digital elevation model may be treated as either a 3D mesh or a 2D texture. However, converting it to a mesh makes it unnecessarily complicated. In addition, the nature and the structure of digital elevation model is different from images or texture. For example, each location in DEM contains a single floating point value while images can contain multiple integer values depending on their type. More importantly, the low-scale characteristics of terrain are different from texture images. Therefore, treating DEM as a 2D texture and using texture synthesizing methods does not necessarily produce suitable result for DEM [6].

There are several works for *Terrain synthesizing* based on DEM [7,14]. However, to the best of our knowledge, there are no other void patching methods that are directly based on DEM and make use of its structure. In addition, our method is context sensitive: it uses the terrain surrounding the void as a guide for creating the patch. In contrast to other model-repair methods, our method uses a multiresolution representation to provide the characteristics for its patches. Thus the novelty of our approach lies in both the use of the DEM structure and the use of contextual, multiresolution characteristics for patching the void.

3 Data Format and Initial Observations

The DEM structure is similar to a regular tensor product surface and allows us to define a simple parameterization of the grid. Usually, it is defined using the two dimensional uv parametric space. The two parameters for the space are denoted with u and v which correspond to the rows and columns of the data.

In general, there are many ways to parameterize a 2D heightmap. The approach described above is popular because the u curves are independent of the v curves. For our purposes, independence is not necessary. The u and v parameters can be described as directions in which to traverse the terrain: they denote the rows and columns. Traversing the terrain using the two diagonal directions will give us two new ways to traverse the terrain (Figure 2(c), 2 (d)) Using these two new directions, we can define an alternative parameterization of the DEM. The new directions are denoted by the w and t parameters respectively (Figure 2(c), 2(d)). Rather, both w and t represent alternative directions in which the characteristics of the surrounding terrain can be captured.

3.1 Observations and General Approach

We approach this problem using the well known divide and conquer method. The first partitioning of the problem are the curve sets for the four parameter directions, each of which can be considered apart from the others. Additionally, each curve set can be further divided into each of the individual curves that compose the set. Considering each curve individually allows us to operate in a 1D parametric set which greatly simplifies the approach. Therefore, as an initial exploration of the problem, we will provide patching methods that individually solves curves.

The set of solutions for a curve set represent an intermediate, but complete, solution for that set. After each of the intermediate solutions for the curve sets are built, they are combined into a final solution. The combination uses a heuristic that prefers samples from solutions that are the most *relevant* to the missing sample. Experimental results show that finding characteristics separately for each curve in a void does not produce the best results.

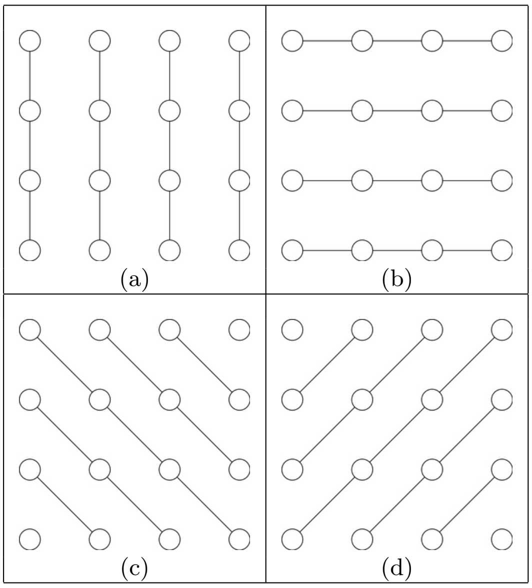


Fig. 2 Illustration of the (a) u curve set for a 4x4 heightmap. (b) and the v curve set for a 4x4 heightmap. (c) and the w curve set for a 4x4 heightmap. (d) and the t curve set for a 4x4 heightmap.

Consequently, we extend the method to find, extract and apply the characteristics to the entire void.

4 Methodology

In this section, we cover our approach for patching the voids found in the SRTM data set. First, we will provide a high level summary of the pipeline. Then we will provide the necessary details of the algorithms that were used. In the case of the characteristics algorithms, we will describe two options that were explored. In the first approach, the characteristics are found individually for each curve in the void. For the final approach, the characteristics are found at once in a single step.

To start, each of the voids in the area are identified. For each void, a smooth patch is built using the overall directions of the surrounding area. Then, the characteristics from the surrounding area are extracted and applied to the smooth patch. In some cases, a number of intermediate solutions are built in this manner. For these cases a heuristic is provided that combines the solutions such that the most relevant results are preferred.

4.1 Reverse Subdivision

Conventionally, wavelets are employed to obtain an efficient and compact representation of the original data [21]. Samavati and Bartels introduced a method for multiresolution curve representation [4] that has local multiresolution filters produced from subdivision methods. In this work we have used multiresolution filters of reverse Chaikin subdivision.

Terrain can be considered to be a tensor-product surface. Subsequently we can apply the reverse subdivision filters to columns, rows or patches of the terrain as in [19]. Denote the terrain at resolution level k as I_j^k , and the corresponding details as D_j^k. A common approach to multiresolution notation is to use a matrix notation where A, B, P, and Q represent the multiresolution operations as follows. The decomposition operations are denoted as: $I^{k-1} = \mathbf{A}^k I^k$, $D^{k-1} = \mathbf{B}^k I^k$. These operations result in a coarse representation I^{k-1} and the corresponding details of the terrain D^{k-1}.

Finally, the reconstruction of the terrain is defined as: $I^k = \mathbf{P}^k I^{k-1} + \mathbf{Q}^k D^{k-1}$. The terrain obtained through the reconstruction process is an exact copy of the original terrain.

4.2 Building The Patch: Curve Case

To provide a solution for an individual curve, the following steps are taken:

1. The boundary of each of the voids is identified (Section 4.3),
2. Using this boundary as a guide, the area surrounding each void is analyzed for its overall direction and a general patch is built to cover the void (Section 4.4),
3. Characteristics of the area surrounding each void are then extracted and applied to the general patch (Section 4.4.1).

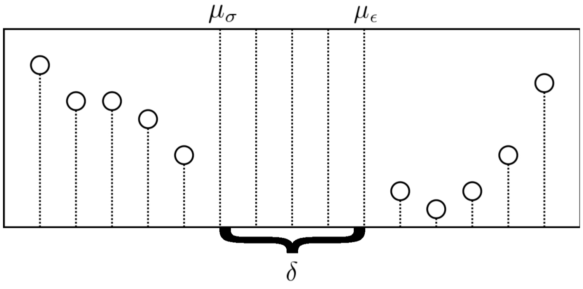


Fig. 3 Illustration demonstrating the boundaries of a void, $\mu_{\sigma+1}$, μ_ϵ and the length of the void: δ.

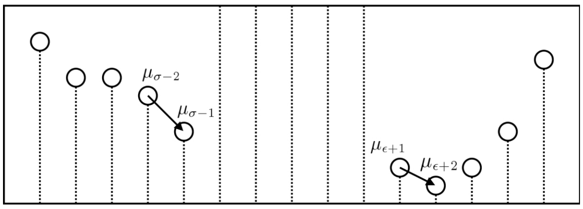


Fig. 4 Illustration of the local terrain direction.

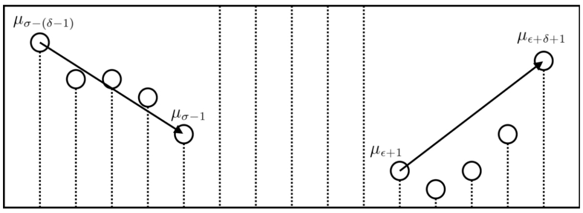


Fig. 5 Illustration of the terrain direction looking in an area the same size as the void.

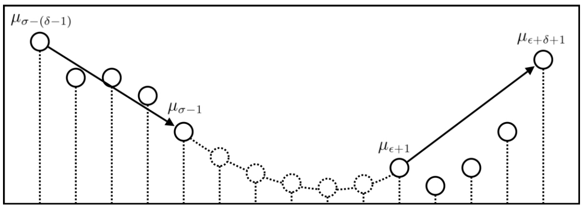


Fig. 6 Illustration of providing a smooth patch using a Hermite curve.

4.3 Finding the Voids: Curve Case

The SRTM data format explicitly marks each missing sample which makes finding voids along a curve straightforward. A search is performed from the beginning of each curve until it finds the first missing sample μ_σ (Figure 3). The end of the void can be found by continuing the search from: $\mu_{\sigma+1}$ until the next non-missing sample $\mu_{\epsilon+1}$ is found. The last missing sample μ_ϵ is marked and the search for the next void continues along the curve from $\mu_{\epsilon+2}$. It is easy to see that the search visits each sample on a curve only once and results in a linear run time method.

4.4 Creating a Smooth Patch: Curve Case

In this step, we create a smooth patch using the general direction of the surrounding terrain. Hermite curves are used to create the patch because they have the same level of continuity G^1 as the Chaikin reverse subdivision (quadratic B-splines) [19,3] that are used for the characteristic extraction. We will generate a single Hermite curve to fill the void. The edges of the surrounding terrain $\mu_{\sigma-1}$ and $\mu_{\epsilon+1}$ are used for the starting and ending points in the Hermite calculation. The last items needed for the Hermite calculation are the tangents at either side of the terrain (Figure 6).

4.4.1 Capturing the Terrain Structure: Curve Case

To produce a patch that closely resembles the surrounding terrain, the direction of the terrain close to the void should be considered. In the curve case, the data on either side of the void will give us the best representation of the direction. An initial idea is to find the tangent of the terrain using the two samples closest to the edge of the void (Figure 4): $\tau_\sigma = \mu_{\sigma-1} - \mu_{\sigma-2}$, and $\tau_\epsilon = \mu_{\epsilon+2} - \mu_{\epsilon+1}$ This choice provides us with the local direction of the terrain near to the edge of the void. In many cases, this local direction is too sensitive to small changes in the terrain (Figure 7). We would rather have the general direction of the terrain at the edge of the void. A possible approach is to use reverse subdivision to reduce the resolution of the surrounding terrain and use its coarse representation. Another approach is to calculate the tangent using a larger area of the surrounding terrain. Either approach will provide a more accurate indication of the general direction of the terrain and will be less sensitive to small changes (Figure 5). In our implementation we use an area that is the same size as the void (Figure 5). This provides us with a better indication of the change in terrain height for areas which are the same size to the void. Let the length of the curve be $\delta = (\mu_\epsilon - \mu_\sigma) + 1$. The tangents on either side of the void can then be found using: $\tau_\sigma = \mu_{\sigma-1} - \mu_{\sigma-(\delta-1)}$, and $\tau_\epsilon = \mu_{\epsilon+\delta+1} - \mu_{\epsilon+1}$.

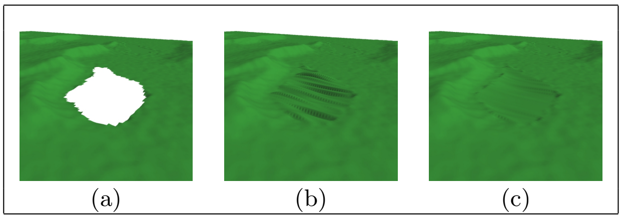


Fig. 7 (a) An example void. (b) Smooth patch for the u curves using the initial calculation of the nearby terrain direction. (c) Smooth patch for the u curves using the larger area for calculating the nearby terrain direction.

4.5 Capturing and Applying Characteristics: Curve Case

The final step in the patching process involves finding and applying characteristics to the void. There are two interesting steps of this method and we shall cover each in turn. First, we discuss how and where the characteristics are found. Second, we discuss how to apply the characteristics to the patch.

A sub-curve from either side of the void is chosen to provide relevant characteristics. These two sub-curves have the same size as the void and are denoted at resolution level k using: μ^k_σ and μ^k_ϵ. Using reverse subdivision for curves, we capture the detail vectors from each of side of the void: ${}^\sigma D^{k-1}$, and ${}^\epsilon D^{k-1}$. Experimental results showed that no more than four levels were needed for most voids encountered in the SRTM dataset, and usually only three are needed. Using the notation from Section 4.1, the extracted characteristics at resolution level i are denoted with: ${}^\sigma D^i = \left[{}^\sigma d^i_1, {}^\sigma d^i_2, \ldots, {}^\sigma d^i_m\right]$, and ${}^\epsilon D^i = \left[{}^\epsilon d^i_1, {}^\epsilon d^i_2, \ldots, {}^\epsilon d^i_m\right]$.

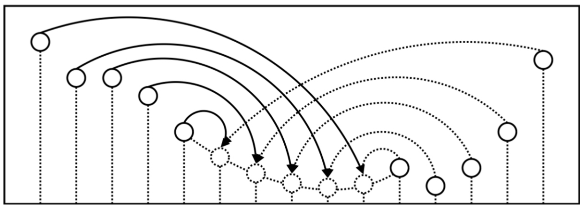


Fig. 8 The source and destination of the extracted characteristics for the mirrored interpolation method.

We now have two sets of characteristics that are near to the void. Before applying the characteristics to the patch, the two sets must be combined in a way that maximizes their relevance to the void. The solution should ensure that the most relevance is given to the characteristics extracted from the curve closest to the respective void edges. In order to ensure a smooth transition between the characteristics on the curve near to the edge of the void and the characteristics on the edge of the patch, a mirroring technique is also employed when combining the characteristics. The technique results in the following characteristic combination at resolution i where m is the index of the last entry in both details vectors:

$$D^i = \begin{bmatrix} \left(\frac{m}{m+1}\right) {}^{\sigma}d^i_m + \left(\frac{1}{m+1}\right) {}^{\epsilon}d^i_m \\ \left(\frac{m-1}{m+1}\right) {}^{\sigma}d^i_{m-1} + \left(\frac{2}{m+1}\right) {}^{\epsilon}d^i_{m-1} \\ \vdots \\ \left(\frac{1}{m+1}\right) {}^{\sigma}d^i_1 + \left(\frac{m}{m+1}\right) {}^{\epsilon}d^i_1 \end{bmatrix}^T . \qquad (1)$$

This final set of characteristics is used to replace all detail vectors in the smooth patch (Figure 8). Essentially, this involves reducing the resolution of the smooth patch, and replacing each level of characteristics with those from Equation 1.

4.6 Patching the Void Using Several Directions

Our experimental results show that applying the characteristics on an individual curve basis is not sufficient. The resulting patches tend to be more smooth than the surrounding terrain. In addition, there is a visible regularity that is introduced with the characteristics. A simple explanation is that the detail vectors found on an individual curve basis may not be compatible. That is, we found that blending the detail vectors from different areas can result in a smoothing of the high frequency information that is present. For these reasons, we investigated extracting characteristics for the entire void from the same 2D area of the terrain, rather than from individual curves. Note that, for this section, we will assume that a smooth patch for the void has already been generated using Hermite curves.

The high-level approach is very similar to the curve case. First, we use a graph traversal algorithm to identify each of the voids. Next, we search the area surrounding the void to find the area that is the best representation of the terrain. Finally, we extract and apply the details using the approach described in Section 4.1 for image based reverse subdivision. The following sections contain the necessary descriptions of the steps for this case.

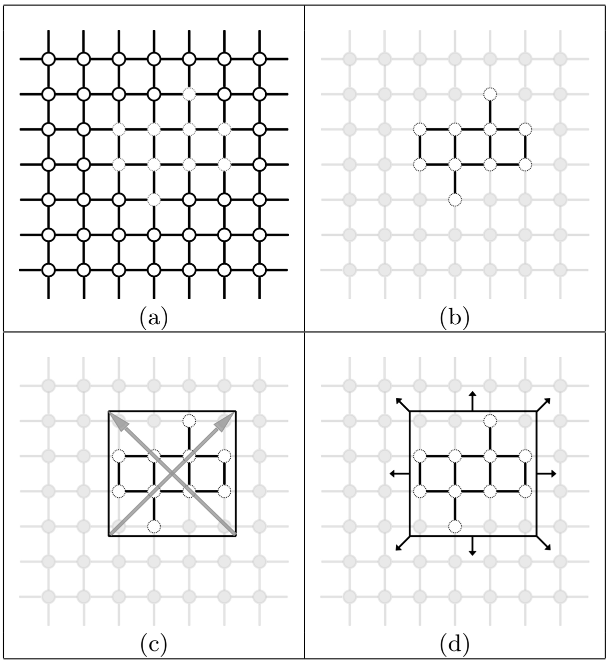


Fig. 9 (a) Heightmap as a connected graph, the dotted circles represent missing samples. (b) A void as a connected component of the graph and (c) its axis aligned bounding box and the vectors used to calculate the normals of the area. (d) The eight possible search directions due to the u, v, w, and t parameter directions.

4.7 Finding the Voids

A heightmap can be represented as a graph where each sample is a vertex of the graph and is directly connected to the samples above, below, to the left and to the right of itself (Figure 9(a)). Using this representation, we can perform a breadth-first graph traversal to identify the missing samples of each void as a connected component (Figure 9(b)). In order to find the first missing sample for each void we iterate over the samples in the heightmap. This sample is given to the graph traversal algorithm which uses it as a starting point.

During the graph traversal, an axis aligned bounding box is created such that it marks the boundaries for each void (Figure 9(c)). It is important to note that the bounding box will not necessarily have dimensions suitable for the reverse subdivision approach [19]. We calculate a target width and height that are suitable for reverse subdivision using the same approach as in

[19]. Then, we make a new bounding box that has the same center as the original box but with a slightly larger width and height. In the following sections we use the new bounding box for both the search and the characteristics extraction.

4.8 Finding the Characteristics

After the boundary for a void has been determined, we find a nearby area that contains the most relevant characteristics for the void. In the 2D case, finding the most relevant information is more challenging than the 1D case as there are more than two directions in which to search. If we consider each of the parameter directions that we use for the curve case, we have eight possible directions in which to search (Figure 9(d)). Each of these areas represents data that is equally relevant to the void, so we must make a choice of which one to use.

To determine which of the candidate areas is most suitable for the void, we use two criteria. The first is that any of the search areas may contain voids making it less suitable. So we count the number of missing samples in that area and use the area with the least missing samples. In some cases, there will be multiple candidate areas that have the minimum number of missing samples. In this situation we prefer the candidate areas that are the same type of terrain as the area surrounding the void. For example, if the void is on the side of a hill, we would like to extract characteristics from the side of a nearby hill. We use a simple geometric comparison to find the best match from the candidate areas. First, we calculate an approximate normal vector for the area that contains the void. Then, we also calculate an approximate normal vector for each of the candidate areas. An approximation of the normal can be calculated using the cross product the two vectors that run diagonally across the bounding box (Figure 9(c)). These normal vectors provide us with the general orientation for each candidate area. We then compare the normals of the candidate areas with the normal for area surrounding the void. The candidate area that has the minimum angle between its normal and that of the void is chosen.

In our experimental results, this normal comparison criteria has proven very effective. However, there are other measurements of terrain, such as local curvature, which might be considered as a criteria for selecting the most compatible candidate area. We leave the exploration of further measures for a future work.

4.9 Applying the Characteristics

At this point, we know the bounding of the void and we have an area chosen which will provide the characteristics for the patch. We then use image based reverse subdivision [19] to extract the coarse approximation from the smooth patch, the details from the target area and combine them to produce the final patch.

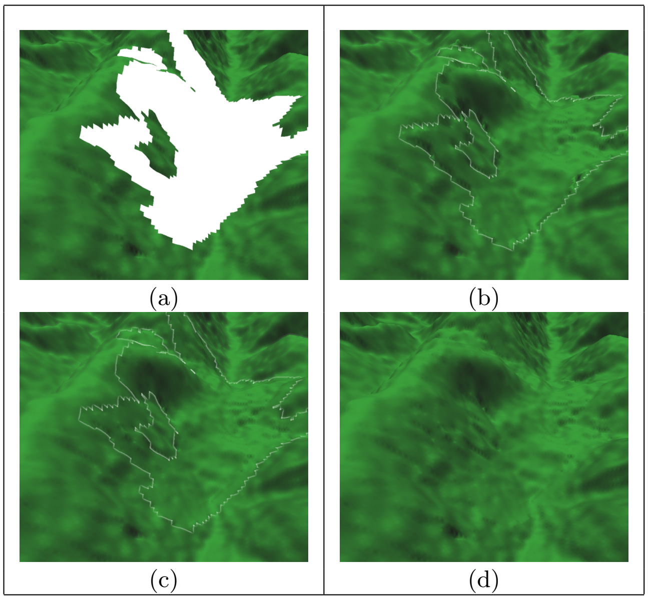


Fig. 10 (a) An example void. (b) Results after patching the void without feathering of the edges with void outline. (c) Results after patching the void with feathering of the edges with void outline. (d) Results after patching the void with feathering of the edges.

The final patch is a rectangular area that fully encompasses the void. However, we should be careful to only use the samples from the patch that are missing in the original void. In addition, there are some cases where the normals at the edges of the void do not match up with the normals at the edges of the patch. The mismatching normals result in a visible discontinuity along the edge of the patched terrain (Figure 10(b)). To minimize these effects, we use a feathering technique for samples on the edge of the void: for each newly generated sample, we first check to see if there is a real sample within a small neighborhood of the point. If there is, an average of the new sample and the surrounding real samples is used (see Figure 10(c) for the effect of this feathering). If there is none, then we use the patched sample unchanged.

4.10 Combining Multiple Solutions and Relevance

Recall from Section 3.1 that we build four intermediate solutions: one for each of the curve directions u, v, w and t. Therefore, we have four possible solutions for each missing sample: one from each curve direction. In the final patch, we can only place one value into the missing sample and we should provide a value that best represents the surrounding terrain. Therefore we combine each of our solutions using a weighted averaging that prefers the more relevant solutions: $\mu_{final} = \sum_{i=1}^{n} \alpha_i \mu_i$ where μ_i

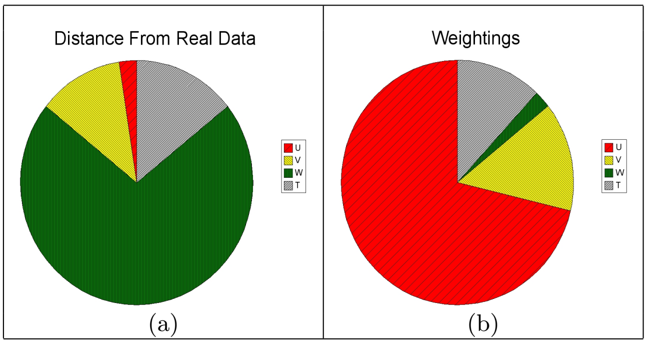


Fig. 11 (a) Pie chart showing distances from real data for a the intermediate solutions for u, v, w, and t curve sets. (b) Pie chart showing the inverse weightings that are associated with the intermediate solutions from u, v, w, and t. Note how the weighting heavily weights the sample that is closest to real data.

represents the solution from curve direction i, and α_i is the weighting for that solution. We only use four terrain directions in this work, however, this equation is general and can be applied for n directions. In addition, the combination should be affine: $\sum_{i=1}^{n} \alpha_i = 1$.

In order to provide the best solution for each missing sample, we should prefer the more *relevant* solution. For each missing sample, assume that we are given a set of intermediate solutions for each curve direction is needed $[\mu_1, \mu_2, \ldots, \mu_n]$. In addition we can find the distance between each missing sample and its closest real sample along that curve direction: $[d_1, d_2, \ldots, d_n]$. Then, we define the total distance as $\delta = \sum_{i=1}^{n} d_i$. Using the original distances and the total distance, we can define a set of inverse ratios, $[r_1, r_2, \ldots, r_n]$ as $r_i = \frac{\delta}{d_i}$, and a new total as: $\Delta = \sum_{i=1}^{n} r_i$. From this, we can easily normalize these ratios thereby defining weights for each of the solution values as: $\alpha_i = \frac{r_i}{\Delta}$.

As a demonstration of the weighting, consider the pie charts in Figure 11. The chart on the left shows the distances between a missing sample and its closest real sample for each curve direction: the largest portion represents the sample that is the least relevant. Conversely, the chart on the left shows the weightings that we have calculated: the largest portion represents the sample that is the most relevant.

5 Conclusions and Experimental Results

In this paper we have presented a novel framework for fixing the voids found in DEMs. The goal is to produce a patch that fits with the surrounding terrain. That is, when presented with the patched terrain, a viewer should not be able to notice the areas that have been patched. The method described provides a smooth patch for voids by analyzing the surrounding terrain in eight directions. In addition, we described a technique that can extract the high-frequency terrain characteristics from the surrounding terrain and apply them to the smooth patch.

Testing the realism of the patches is achieved by viewing the patches. We present the results of five tests on terrain that varies in both the general steepness and overall characteristics. They range from relatively flat areas to mountainous areas (Figure 12)). In all of these cases, the patches that are produced are an excellent match for the surrounding terrain. The areas of the void are undetectable without the aid of the photos that show the voids.

Statistical measures of terrain are typically related to synthesis methods. As we discussed, in our method the patches are generated in such a way that the variations from the base curve follow the same measures as the surrounding terrain. Therefore, our method inherently uses basic statistical plausibility to fill the voids, however we also solve the issue of visual artifacts. In addition, visual artifacts may not affect statistical evaluation of the patch but are detrimental to visual quality. Thus, the visual inspection process provides the main validation of our patches.

Terrain does not have C^1 continuity and therefore may have large features in small areas. We cannot reproduce these features in areas where there is no evidence of the feature in the surrounding terrain. However, to provide a validation of our work, we have created an artificial void in an area of terrain where we have complete data. This allows us to compare our synthetic results with the real terrain to determine if our patch is indeed realistic, see Figure 13. The patches in both illustrations have characteristics that fit with the surrounding terrain and fit with the original data for the void.

References

1. Ashikhmin, M.: Synthesizing natural textures. In: SI3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics, pp. 217–226. ACM Press, New York, NY, USA (2001)
2. Baraquet, G., Sharir, M.: Filling gaps in the boundary of a polyhedron. Computer Aided Geometric Design **12**(2), 207–229 (1995)
3. Bartels, R.H., Beatty, J., Barsky, B.: An Introduction to Splines for use in Computer Graphics and Geometric Modeling. Morgan Kauffman (1987)
4. Bartels, R.H., Samavati, F.F.: Reversing subdivision rules: Local linear conditions and observations on inner products. Journal of Computational and Applied Mathematics **119**(1-2), 29–67 (2000)
5. Bonet, J.S.D.: Multiresolution sampling procedure for analysis and synthesis of texture images. In: Computer Graphics, pp. 361–368. ACM SIGGRAPH (1997). URL `http://www.debonet.com/Research/TextureSynthesis`
6. Brosz, J.: Terrain modeling by example. Master's thesis, University of Calgary, Calgary, Alberta (2005)

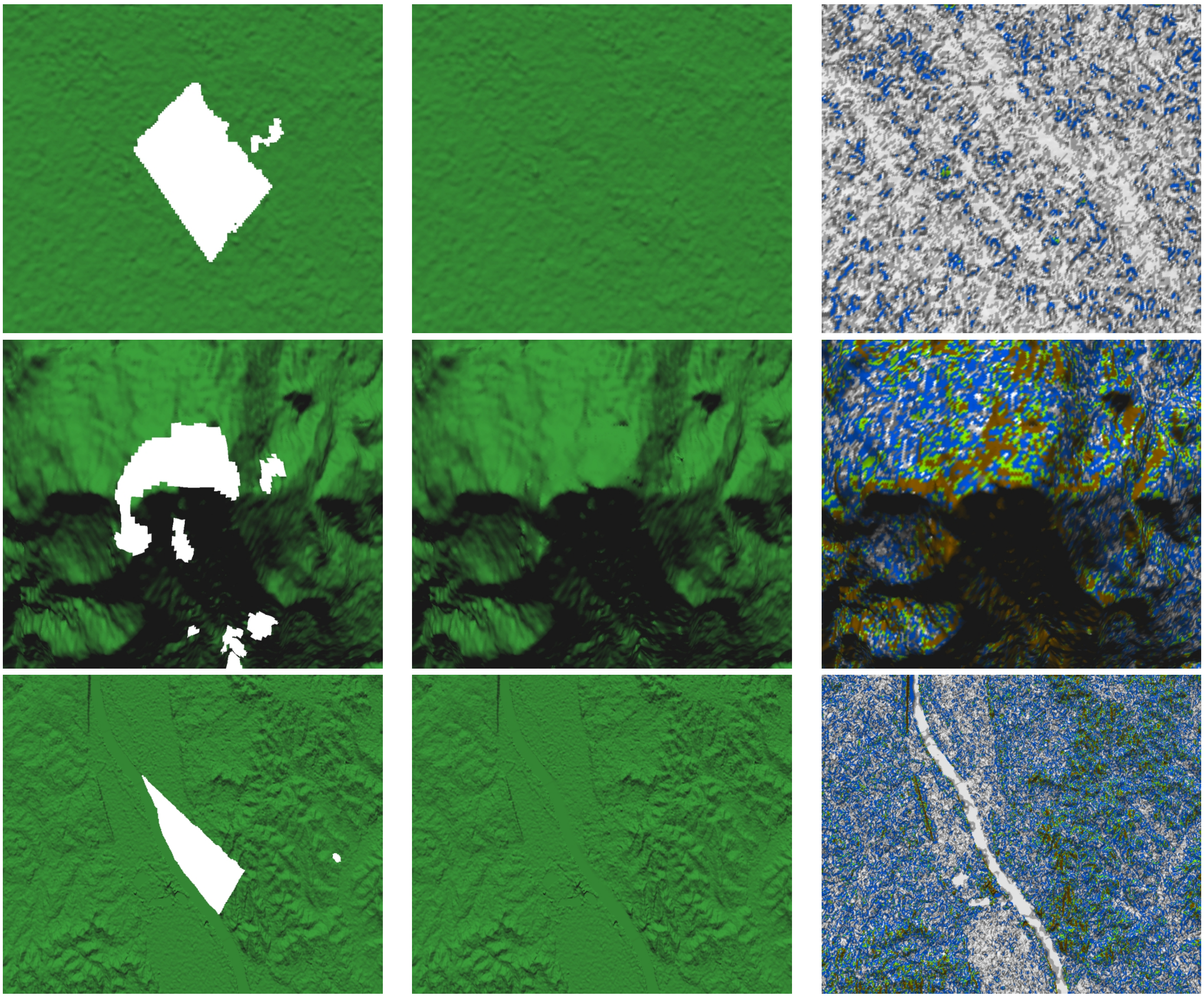


Fig. 12 The left column shows the void, the middle column the patch with smooth shading, and the right column shows a custom rendering using slope steepness [6] of the details [24].

7. Brosz, J., Samavati, F.F., Costa Sousa, M.: Terrain synthesis by-example. In: International Conference on Computer Graphics Theory and Applications. In Association with Eurographics (2006)
8. Carr, J.C., Beatson, R.K., Cherrie, J.B., Mitchell, T.J., Fright, W.R., McCallum, B.C., Evans, T.R.: Reconstruction and representation of 3d objects with radial basis functions. In: SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pp. 67–76. ACM Press (2001)
9. Cohen, M.F., Shade, J., Hiller, S., Deussen, O.: Wang tiles for image and texture generation. Proceedsing of SIGGRAPH 2003 (2003)
10. Davis, J., Marschner, S., Garr, M., Levoy, M.: Filling holes in complex surfaces using volumetric diffusion. In: First International Symposium on 3D Data Processing Visualization and Transmission, pp. 428–861 (2002)
11. Drori, I., Cohen-Or, D., Yeshurun, H.: Fragment-based image completion. ACM Trans. Graph. **22**(3), 303–312 (2003)
12. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. Proceedings of SIGGRAPH 2001 pp. 341–346 (2001)
13. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: IEEE International Conference on Computer Vision, pp. 1033–1038. Corfu, Greece (1999)
14. Fournier, A., Fussell, D., Carpenter, L.: Computer rendering of stochastic models. Commun. ACM **25**(6) (1982)
15. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. In: SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pp. 327–340. ACM Press, New York, NY, USA (2001)
16. Li, B., Qi, Y., Shen, X.: An image inpainting method. Computer Aided Design and Computer Graphics, 2005. Ninth International Conference on pp. 6 pp.– (7-10 Dec. 2005)
17. Liang, L., Liu, C., Xu, Y., Guo, B., Shum, H.: Real-time texture synthesis by patch-based sampling. ACM Trans. Graph. **20**(3), 127–150 (2001)
18. Liepa, P.: Filling holes in meshes. In: SGP '03: Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing, pp. 200–205. Eurographics Association (2003)

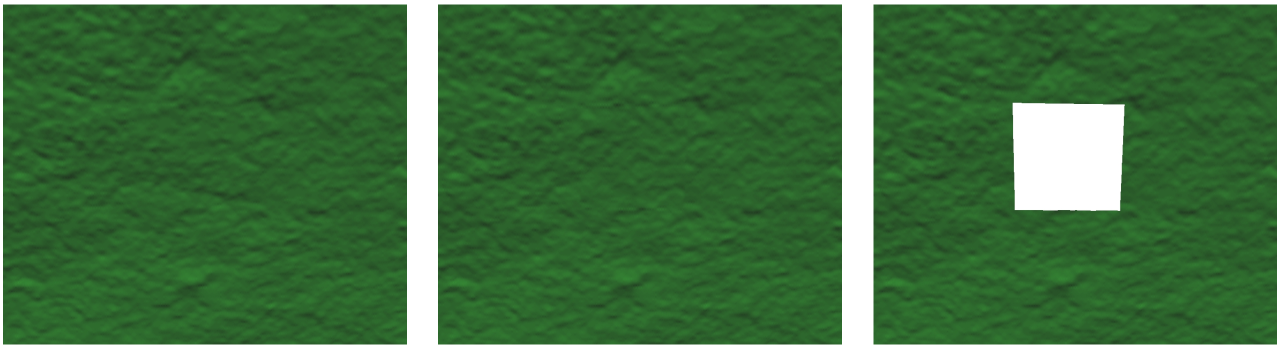
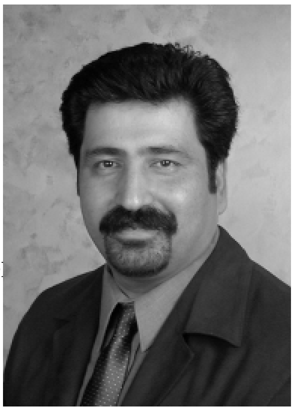


Fig. 13 Illustrates results of our patch for voids for which we have real data. On the left is the original terrain, our patch is in the middle and the void is on the right.

19. Samavati, F.F., Bartels, R.H., Olsen, L.: Local b-spline multiresolution with examples in iris synthesis and volumetric rendering. In: Synthesis and Analysis in Biometrics. World Scientific Publishing (2006)
20. Sharf, A., Alexa, M., Cohen-Or, D.: Context-based surface completion. ACM Trans. Graph. **23**(3), 878–887 (2004)
21. Stollinitz, E., Derose, T., Salesin, D.: Wavelets for Computer Graphics. Morgan Kaufmann, San Francisco (1996)
22. USGS: Shuttle radar topography mission (srtm) faq. http://seamless.usgs.gov/website/seamless/faq/srtm_faq.as
23. Verdera, J., Caselles, V., Bertalmio, M., Sapiro, G.: Inpainting surface holes. Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on **2**, II–903–6 vol.3 (14-17 Sept. 2003)
24. Wecker, L.: Synthesizing techniques based on multiresolution. Master's thesis, University of Calgary, Calgary, Alberta (2007)
25. Wei, L., Levoy, M.: Fast texture synthesis using tree-structured vector quantization. In: SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pp. 479–488. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA (2000)
26. Zelinka, S., Garland, M.: Jump map-based interactive texture synthesis. ACM Trans. Graph. **23**(4), 930–962 (2004)

Lakin C. Wecker is a senior software developer for Evolvs Media Inc. in Calgary AB. He has over 6 years experience providing application development, Internet solutions, and systems administration for commercial clients and open source projects. He will receive his MSc degree from the University of Calgary in the Summer of 2007. His research interests include Computer Graphics, Multiresolution, and Visualization. He has authored two papers in Multiresolution Synthesis Techniques.



Faramarz F. Samavati is an associate professor in the Department of Computer Science, University of Calgary. He is also an adjunct associate professor in Computer Engineering Department, Technical University of Lisbon, Portugal. He received his PhD degree from Sharif University of Technology in 1999. He was a research visitor at University of waterloo in 1997. Prof. Samavatis research interests are Computer Graphics, Geometric Modeling, Visualization, and Computer Vision. He has authored more than 40 research papers in Subdivision Surfaces, Sketch based Modeling, Multiresolution and Wavelets, Surface Modeling and Scientific Visualization.

Marina L. Gavrilova is an Associate Professor in the Department of Computer Science, University of Calgary. Dr. Gavrilova's research interests include computational geometry, image processing, optimization, exact computation and computer modelling. Prof. Gavrilova is a founder of two innovative research labs, the SPARCS Laboratory for Spatial Analysis in Computational Sciences and the Biometric Technologies Laboratory. Her publication list includes over 80 research papers, books and book chapters. Prof. Gavrilova founded the annual International Workshops on Computational Geometry and Applications series in 2001, and in 2003 Co-Founded the first ICCSA International Conference on Computational Science and Applications, now in its fifth edition. Prof. Gavrilova is an Editor-in-Chief for Transactions on Computational Science Journal series (Springer) and serves on the Editorial Board for International Journal of Computational Sciences and Engineering and Computer Graphics and CAD/CAM Journal. She is an ACM, IEEE and Computer Society member.